An Algorithm For Classification Of Multi-Spectral Data And Its Implementation On A Massively Parallel Computer*

Behzad M. Shahshahani, David A. Landgrebe
School Of Electrical Engineering
Purdue University
West Lafayette, IN 47906

Tel: (317) 494-1743 Fax: (317) 494-6440
behzad@ecn.purdue.edu landgreb@ecn.purdue.edu

Abstract

A new method for classification of multi-spectral data is proposed. This method is based on fitting mixtures of multivariate Gaussian components to training and unlabeled samples by using the EM algorithm. Through a backtracking search strategy with appropriate depth bounds , a series of mixture models are compared. The validity of the candidate models are evaluated by considering their description lengths and allocation rates. The most suitable model is selected and the multi-spectral data are classified accordingly. The EM algorithm is mapped onto a massively parallel computer system to reduce the computational cost. Experimental results show that the proposed algorithm is more robust against variations in training samples than the conventional supervised Gaussian maximum likelihood classifier.

1. Introduction

The conventional method for classification of multi-spectral remote sensing data involves the estimation of class statistics from training samples and plugging the estimates into quadratic decision rules. This method is very sensitive to the choice of training samples especially when their number is small. Often, the training samples are not good representatives of the true class populations, and sometimes training samples from one or more classes are not available. In addition, the Gaussian assumption about a class is sometimes incorrect and the analyst has to select sub-classes which can be modeled with Gaussian distributions.

The unlabeled samples, which are available in large numbers, may provide the additional information needed to design suitable classifiers. It has already been shown that the small sample size problem can be mitigated by using unlabeled samples[1]. It may also be possible to correct the estimates of class statistics, which might have been incorrectly estimated due to unrepresentative training samples, by incorporating unlabeled samples. In this paper, we describe a method for classifying multi-spectral data that is based on both training and unlabeled samples. This method is based on fitting mixture densities to available data. Various mixture models are examined through a backtracking search strategy and the best model in terms of clustering ability and description length is selected. The data are then classified according to the selected model. The EM mixture identification procedure is mapped onto a massively parallel computer system.

2. Fitting Normal Mixtures to Data

Consider a problem involving J classes, denoted by S_j (j=1,...,J), where each class consists of one or several Gaussian components. Denote the total number of Gaussian components by m. The pdf p(x) can be written as a mixture of these m Gaussian components:

$$p(x|\Phi) = \sum_{i=1}^{m} \alpha_i p_i(x|\phi_i) \quad , \qquad x \in R^d \tag{1}$$

where

$$\sum_{i=1}^{m} \alpha_i = 1 \qquad \alpha_i > 0$$

* SPIE International Symposium on Optical Applied Science and Engineering., July 11-July 16, 1993 San Diego, CA. This work was supported in part by NASA under Grant NAGW-925. A MasPar computer system used in this work is maintained and operated by the Parallel Processing Laboratory of the School of Electrical Engineering, Purdue University and is supported by NSF Parallel Infrastructure Grant CDA-9015696.

$$\phi_i = (\mu_i, \Sigma_i), \qquad \Phi = (\alpha_1, \cdots, \alpha_m, \mu_1, \cdots, \mu_m, \cdots, \Sigma_1, \cdots, \Sigma_m)$$

α_i is the prior probability of the i^{th} component and μ_i and Σ_i are its mean vector and covariance matrix. The maximum likelihood estimates of the parameters of the mixture in the presence of N unlabeled samples $x_1,...,x_N$ drawn randomly from the mixture and N_j training samples from each class j denoted by $z_{j1},...,z_{jN_j}$ can be found iteratively through the use of the following EM (Expectation-Maximization) equations[2]:

For $i \in S_j$

$$\alpha_i^+ = \frac{\sum_{k=1}^{N_j} P_j^c(i|z_{jk}) + \sum_{k=1}^{N} P^c(i|x_k)}{N\left(1 + \frac{N_j}{\sum_{r \in S_j} \sum_{k=1}^{N} P^c(r|x_k)}\right)} \tag{2a}$$

$$\mu_i^+ = \frac{\sum_{k=1}^{N_j} P_j^c(i|z_{jk}) z_{jk} + \sum_{k=1}^{N} P^c(i|x_k) x_k}{\sum_{k=1}^{N_j} P_j^c(i|z_{jk}) + \sum_{k=1}^{N} P^c(i|x_k)} \tag{2b}$$

$$\Sigma_i^+ = \frac{\sum_{k=1}^{N_j} P_j^c(i|z_{jk})(z_{jk} - \mu_i^+)(z_{jk} - \mu_i^+)^T + \sum_{k=1}^{N} P^c(i|x_k)(x_k - \mu_i^+)(x_k - \mu_i^+)^T}{\sum_{k=1}^{N_j} P_j^c(i|z_{jk}) + \sum_{k=1}^{N} P^c(i|x_k)} \tag{2c}$$

In the above equations superscripts + and c denote the next and current values of the parameters respectively and:

$$P_j^c(i|z_{jk}) = \frac{\alpha_i^c p_i(z_{jk}|\phi_i^c)}{\sum_{r \in S_j} \alpha_r^c p_r(z_{jk}|\phi_r^c)} \quad \text{and} \quad P^c(i|x_k) = \frac{\alpha_i^c p_i(x_k|\phi_i^c)}{p(x_k|\Phi^c)}$$

Starting from any reasonable point in the parameter space a maximum of the likelihood function can be found by iteratively using the above equations[3].

Sometimes it may be desirable to constrain the prior probability of one (or more) of the classes to be smaller (or larger) than a predefined value. For example in segmentation of agricultural fields, in addition to the possible crop types, it may be desirable to consider an additional class for representing urban areas, roads and other non vegetation ground covers. In such a case, it may be helpful to constrain the prior probability of this class to be smaller than a predefined number ε that reflects the prior knowledge of the analyst about the maximum percentage of such ground cover. This prevents the possibility that bad initial parameter values will cause the estimation process to result in an unreasonably high prior probability for the non-vegetation class. Constrained maximum likelihood estimation[4] should be used for this purpose. Let us assume that the following additional constraint is imposed in the maximum likelihood estimation:

$$\sum_{i \in S_1} \alpha_i < \varepsilon \tag{3}$$

If the EM algorithm is used for estimating the parameters values, the above constraint should be imposed in the Maximization part of the process. By considering the Kuhn-Tucker conditions, it is easy to see that the constrained EM algorithm is described by the following process:

1) Perform one iteration of the regular EM algorithm.
2) Check the constraint (3). If it is satisfied go to step 1, otherwise go to step 3.

3) For all components $i \in S_1$ update the prior probabilities as follows:

$$\alpha_i^+ \leftarrow \frac{\alpha_i^+}{\sum_{r \in S_1} \alpha_r^+} . \varepsilon$$

For all components $i \notin S_1$ update the prior probabilities as follows:

$$\alpha_i^+ \leftarrow \frac{\alpha_i^+}{\sum_{r \in S_1} \alpha_r^+} . (1 - \varepsilon)$$

Go to step 1.

Additional constraints on more classes can also be imposed in a manner similar to that described in reference (4).

The EM process can be easily mapped onto a parallel processor as will be discussed in section 5.

3. Model Evaluation

After a particular mixture model is considered for p(x) and its parameters are estimated, its validity needs to be evaluated. There has been much work done on methods for evaluating the validity of a model for clustering data into groups. Some of these methods are summarized in reference (5). We describe two methods in this section and use them in section 4 for selecting a suitable mixture model for the pdf of the samples.

Minimum Description Length (MDL)

Rissanen[6] proposed the MDL method for model selection. Since then it has been used in variety of applications. Examples of related works are references (7) and (8). According to MDL, the best model is the one which yields the shortest code for the data. The length of the shortest code for the data can be obtained, from Shannon's theorem, as a function of the probability model that underlies the generation of the data. Different probability models obtain different code lengths. Therefore models can be compared based on the codes that they generate for the data. However, as one increases the complexity of a probability model, higher likelihood for the data can be obtained, and therefore shorter codes can be constructed. The MDL principle penalizes the more complex probability models. The argument used in the MDL method is that a probability model that is used for encoding must be known at the time of decoding the data in order to create the necessary *code book*. Therefore, in a hypothetical transmission system, the parameters of the probability model need to be transmitted prior to transmission of the encoded data. Hence, the total code length is the summation of the code length for the data plus that for the model parameters. Now, let's consider a mixture model such as the one in equation (1). According to Shannon's theorem, under this pdf model, the shortest prefix code length for the N independent unlabeled samples $x_1,...,x_N$ is the following:

$$L(X|Model) = -\sum_{k=1}^{N} \log p(x_k|\Phi) \tag{4}$$

where L denotes the code length. The code length for the N_j training samples from class S_j is:

$$L(Z_j|Model) = = -\sum_{k=1}^{N_j} \log \left(\sum_{r \in S_j} \frac{\alpha_r}{\sum_{t \in S_j} \alpha_t} p_r(z_{jk}|\phi_r) \right) \tag{5}$$

The total code length for transmitting all the unlabeled and training samples is therefore the following:

$$L(X,Z|Model) = -\sum_{k=1}^{N} \log p(x_k|\Phi) - \sum_{j=1}^{J} \sum_{k=1}^{N_j} \log \left(\frac{1}{\sum_{t \in S_j} \alpha_t} \sum_{r \in S_j} \alpha_r p_r(z_{jk}|\phi_r) \right) \tag{6}$$

Before the codes for the data are transmitted, the model parameters need to be sent to receiver. The receiver can then use the model parameters to construct the codes for the upcoming data. In the absence of knowledge about the prior distribution of the model parameters, Rissanen proposed the use of a universal prior according to which the code length for transmitting the parameters of the model is :

$$L(\text{Model}) = -\log p(\Phi) = \frac{1}{2} k \log N^* \tag{7}$$

Here, N^* denotes the total number of samples (training and unlabeled in our case), k is the number of free parameters in the model. For a normal mixture density such as the one in equation (3) one can write:

$$k = m\left(d + \frac{d(d+1)}{2} + 1\right) - 1$$

where d is the dimension of the feature space. Therefore the total code length for transmitting the unlabeled samples, training samples and parameter values is:

$$L(X,Z,\text{Model}) = -\sum_{k=1}^{N} \log p(x_k|\Phi) - \sum_{j=1}^{J}\sum_{k=1}^{N_j} \log\left(\frac{1}{\sum_{t\in S_j}\alpha_t}\sum_{r\in S_j}\alpha_r p_r(z_{jk}|\phi_r)\right) + \frac{1}{2} k \log N^* \tag{8}$$

The parameter set Φ for each model under consideration should be estimated so that the above is minimized. To do so, the first two terms of L(X,Z,Model) have to be minimized with respect to Φ. The solution is the maximum likelihood estimate of Φ which can be found by the EM equations of section 2. Among a set of competing models, the one which has the smallest description length (according to equation (8)), should be selected.

Total Allocation Rate

After a model is selected for p(x), and its parameters are estimated, the samples are classified according to their maximum a posterior probabilities; each sample is classified to the class for which this probability is highest. The accuracy of the classification can be estimated by using the training samples. Let $I_j(x)$ be the indicator function for class S_j: it is one if x is classified to class S_j based on its posterior probability and is zero otherwise. The correct allocation rate (percentage of correctly classified samples) for class S_j can be estimated by the count estimator:

$$C_j = \frac{\sum_{k=1}^{N_j} I_j(z_{jk})}{N_j} \tag{9}$$

The total correct allocation rate is therefore :

$$C = \sum_{j=1}^{J}\left(\sum_{i\in S_j}\alpha_i\right) C_j \tag{10}$$

The above estimator for the total correct allocation rate needs training samples from all classes. Alternatively, unlabeled samples can be used to estimate the allocation rates. In references (9,10), posterior probabilities of unlabeled samples are used for estimating the probability of correct classification. Based on these ideas, in reference (11) the use of the estimates of the allocation rates using unlabeled samples for cluster analysis is studied. Intuitively, one can argue that if the unlabeled samples are classified based on maximum a posterior probabilities, then a good mixture model is one under which all of these posterior probabilities have values near one. In other words, the clustering is not very fuzzy since each sample is classified with high probability. The estimates of class conditional and total allocation rates based on unlabeled samples are[11]:

$$C_j = \frac{\sum_{k=1}^{N} \max_r P(S_r|x_k) I_j(x_k)}{N \sum_{i \in S_j} \alpha_i} \tag{11}$$

$$C = \frac{\sum_{k=1}^{N} \max_r P(S_r|x_k)}{N} \tag{12}$$

In our case of interest, both training and unlabeled samples are available. Therefore, we use the following estimates for the allocation rates that are based on both types of samples:

$$C_j = \frac{\sum_{k=1}^{N_j} I_j(z_{jk}) + \sum_{k=1}^{N} \max_r P(S_r|x_k) I_j(x_k)}{N_j + N(\sum_{i \in S_j} \alpha_i)} \tag{13}$$

$$C = \sum_{j=1}^{J} \left(\sum_{i \in S_j} \alpha_i \right) C_j \tag{14}$$

We use equation (14) and refer to it as TAR (Total Allocation Rate), for evaluating different mixtures models. If TAR is near one, it means that under the proposed model the training samples are classified correctly and unlabeled samples are classified with high posterior probabilities. Notice that the these estimates are all optimistically biased for the true allocation rates since the samples are already used for estimating the parameters. The bootstrap method can be used for bias correction[11].

4. Model Selection By Search

Using the model validity criteria described in section 3, one can select a particular normal mixture model for p(x) from a set of such models. A backtracking search strategy is utilized here for this purpose. The process starts from an initial normal mixture model. The parameters of the model under consideration are estimated using the EM algorithm of section 2 that is based on both training and unlabeled samples. Next, the model is evaluated and a new model is derived by adding a new Gaussian component through splitting one of the current components. If the new model is more suitable than its parent model, the search continues from the new model, otherwise it backtracks to the parent model and another component is split. The process continues until either a suitable model is found, or the search ends by reaching the depth bounds and the most suitable model is then selected. The overall process is described below.

```
main program:
                BackTrack ( Initial Model )

BackTrack (Model):
                if Model is Accepted  save Model,    return (Found)
                if Depth Bound        return(Fail)
        Loop    if (no components left unsplit in Model)  return(Fail)
                split next component in the split candidate list
                identify the NewModel
                out = BackTrack (NewModel)
                if (out = Fail)   goto Loop
                else return(Found)
```

The initial model can possibly be a normal mixture density in which each informational class (a class from which training samples are gathered and therefore are of information to the user) has a single component. Alternatively the training samples from each informational class can be used to find an initial mixture model for each class as described in reference (2). An additional class (*unknown* class) can also be considered to take care of possible ground covers from which training samples are not available (such as farmsteads and roads in an agricultural area). The initial parameter values for the EM algorithm are

estimated from the training samples. For the *unknown* class some of the unlabeled samples can be used (e.g. 5% of the furthest unlabeled samples from the informational classes) to estimate the initial parameter values. A small number ε (e.g. 0.1) may be selected for the maximum prior of the *unknown* class.

The process ends by accepting a model whose TAR value is above a threshold. We have used 95% threshold in our experiments.

A new model is generated by splitting a component into two and re-estimating the parameter values using the EM formulas. A component is split into two new components along its principal axis. The child components have equal initial prior probabilities, equal covariance matrices (equal to 0.25 times the covariance of the parent component), and means at equal distances from the mean of the parent component along its principal axis.

For each model under consideration, the components are placed and ranked in a split candidate list. The ranking of the components may be based on their contribution to the misclassification of the training samples.

Several depth bounds are used to constrain the search space. The first depth bound is based on MDL. If a new model has a larger description length than its parent it is rejected and the process backtracks. The second depth bound is reached when a new model has a lower TAR than its parent. An additional depth bound is placed on the number of components in an informational class. Since the association of the components to classes is based on training samples and it is unreasonable to think that a handful of training samples can identify a large number of components, we limit the maximum number of components for an informational class as a function of the number of training samples from that class. Intuitively, we argue that at least d+1 training samples are required to identify a component as a member of a class. Therefore, the maximum number of components per informational class S_j is $\lfloor N_j / (d+1) \rfloor$ (minimum is of course 1). In this way, on average d+1 training samples belong to each component of an informational class. This prevents the covariance matrix of the components from ever becoming singular and therefore reduces the chance of the EM algorithm to fall into singular points of the parameter space.

If the process ends without finding a suitable model with TAR above the pre-specified threshold value, then among all the models that were not rejected by the depth bounds the one with maximum TAR is selected and ties are broken by choosing the one that has the minimum description length.

5. Parallel Implementation on a Massively Parallel Computer

The estimation of the parameters of a normal mixture density through the iterative EM equations of section 2 can be mapped onto a parallel computer system. The computational complexity per iteration is $O(mN^*d^2)$ on a serial machine (N^* being the total number of training and unlabeled samples). The computations that need to be performed on each sample can be done separately from others. Therefore, the samples can be distributed among the processing elements (PE's) of a parallel computer to get a per iteration computational complexity of $O(m(N^* \text{ Mod } nproc)d^2)$ where *nproc* is the number of processors. In a massively parallel computer *nproc* is very large and hence N^* Mod *nproc* is small. We have used a MasPar MP-1 computer system with 16,384 processing elements in our implementation of the algorithm. Some bench mark results on the cpu time spent for 10 iterations of the EM algorithm are listed in Table 1. The numbers in this table are averages over 5 independent trials. We do not claim that the codes used are optimal and in fact, further improvement on the parallel version of the program is most likely possible.

Table1: cpu times spent per 10 iterations of the EM algorithm on a MasPar MP-1, Titan Ardent P3, and a Sun 4 Sparc. Numbers in parentheses indicate the ratio between the serial machine cpu time and that of the MasPar. Programs are written in C for both serial machines and in MPL-C for the MasPar. If an optimizing compiler is used for the Sparc a reduction of about 35-40 % in cpu time is observed.

dimension	# components	# samples	cpu time (sec) MasPar MP1	cpu time (sec) Ardent P3	cpu time (sec) Sun 4 Sparc
4	2	3000	0.90	4.46 (4.96)	12.59 (13.99)
4	2	16384	0.89	27.69 (31.11)	69.67 (78.28)
4	4	3000	1.72	8.47 (4.92)	24.47 (14.22)
4	4	16384	1.72	53.48 (31.09)	138.37 (80.45)
7	2	3000	2.77	9.51 (3.43)	28.72 (10.37)
7	2	16384	2.77	58.94 (21.28)	160.43 (57.92)
7	4	3000	5.48	18.82 (3.43)	59.76 (10.91)
7	4	16384	5.48	118.22 (21.57)	321.28 (58.63)

6. Experimental Results

The experiments in this section are performed on Landsat Thematic Mapper (TM) data taken over Tippecanoe County, Indiana in July 1986. Ground truth data was gathered at the same time. Four informational classes are selected in the scene: corn, soybean, wheat and alfalfa/oats. From the ground truth map, the numbers of pixels from the informational classes are found to be: 9371, 8455, 1923 and 2175 respectively. All of the seven bands are used in the experiments.

Two different experiments are performed with two different sets of training samples. In experiment 1, training samples are selected very carefully by an analyst through a trial and error procedure in order to maintain a high classification accuracy when a Gaussian maximum likelihood (GML) classifier is used. These training fields are highlighted in figure 1 (a). The total classification error of the GML classifier is 19.98 %. Figure 1(b) shows the error map for the GML classifier. Dark pixels in the error map indicate the pixels that were classified incorrectly. Using these *good* training fields and 1849 unlabeled samples drawn from the image (every fourth pixel in every fourth line), the proposed algorithm achieved a classification error of 18.37 %. An extra class was used as the *unknown* class and its prior probability was constrained to less than 0.1. The error map for this case is illustrated in figure 1 (c). The same procedure was performed with 3249 unlabeled samples (every third pixel in every third line). The classification error was 18.46%. Figure 1(d) shows the error map for this case.

Next, the training fields were altered in order to illustrate the result of using small unrepresentative training samples in the classification process. The new training fields are high lighted in figure 2(a). The GML classifier achieved a classification error of 40.63% in this case; a 20.65% increase in error from the last experiment. Figure 2(b) shows the error map for the GML classifier. Using 1849 unlabeled samples, the proposed method achieved 17.85% classification error and using 3249 unlabeled samples the classification error was further reduced to 15.38%. Figures 2(c) and 2(d) illustrate the error maps for these two cases respectively. Obviously, the GML classifier is very sensitive to the number and quality of training samples. This sensitivity was reduced by incorporating the unlabeled samples through the proposed algorithm.

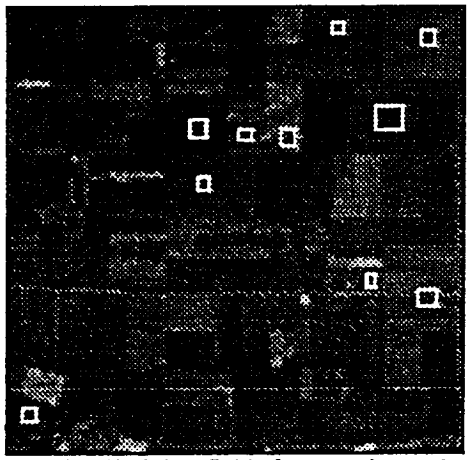

Fig 1(a): Training fields for experiment 1.

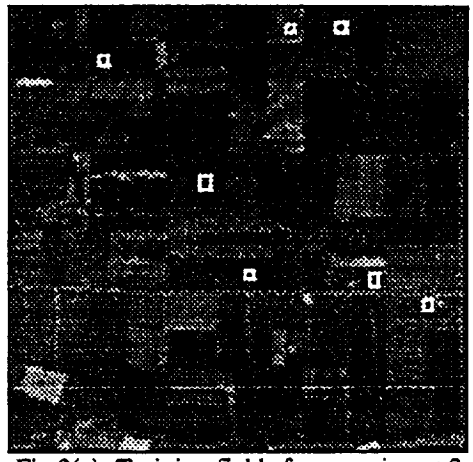

Fig 2(a): Training fields for experiment 2.

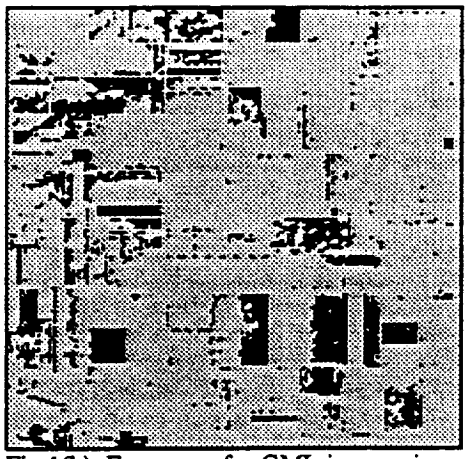

Fig 1(b): Error map for GML in experiment 1. Error = 19.98%

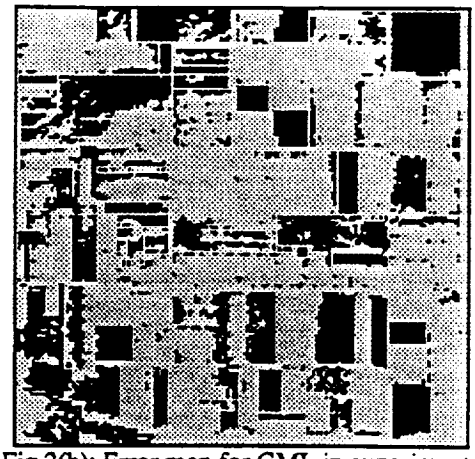

Fig 2(b): Error map for GML in experiment 2. Error = 40.63%

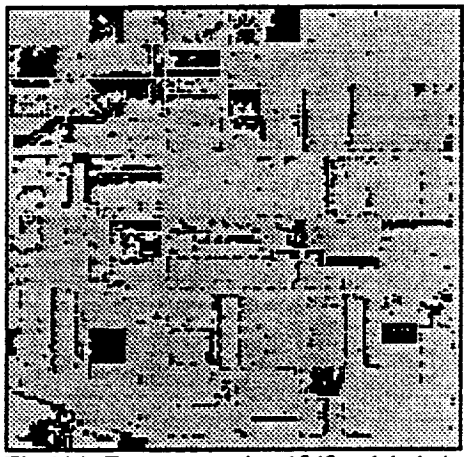

Fig 1(c): Error map using 1849 unlabeled samples in experiment 1. Error = 18.37%

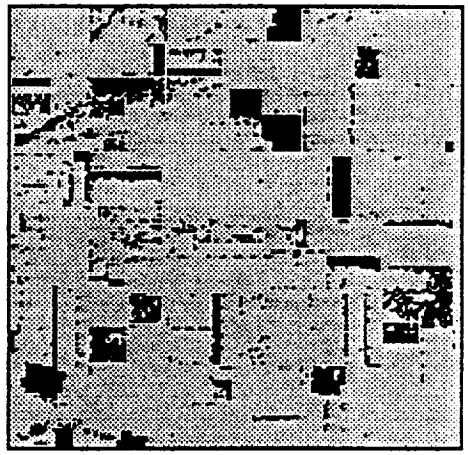

Fig 2(c): Error map using 1849 unlabeled samples in experiment 2. Error = 17.85%

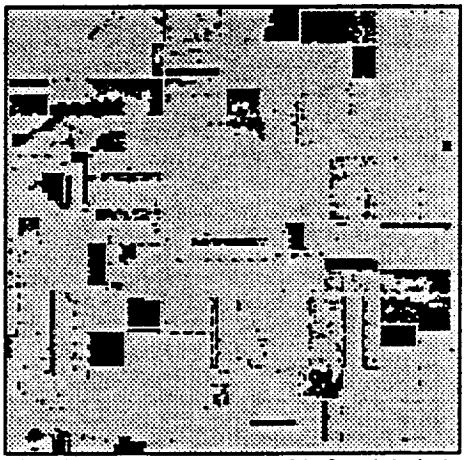

Fig 1(d): Error map using 3249 unlabeled samples in experiment 1. Error = 18.46%

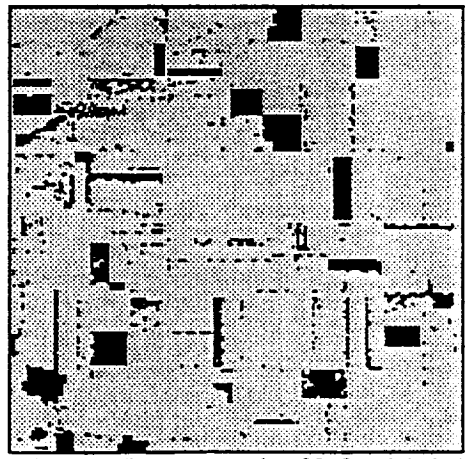

Fig 2(d): Error map using 3249 unlabeled samples in experiment 2. Error = 15.38%

7. Concluding Remarks

A method for classification of multi-spectral data was proposed. This method is based on using both training and unlabeled samples for estimating a suitable model for the probability density function of the feature space. In this way, the proposed method reduces the sensitivity of the decision rule with respect to small unrepresentative training samples. The implementation of the algorithm on MasPar computer system was discussed. Experiments on real data show encouraging results.

8. References

1 B.M. Shahshahani, D.A. Landgrebe, "Use of Unlabeled Samples for Mitigating the Hughes Phenomenon," *Proc. Int. Geoscience & Remote Sensing Symp IGARSS'93*, Tokyo, Japan August 1993

2 B.M. Shahshahani , D.A. Landgrebe, "Using Partially Labeled Data for Normal Mixture Identification with Application to Class Definition," *Proc. IEEE International Geoscience & Remote Sensing Symposium*, pp 1603-1605, 1992

3 A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood Estimation From Incomplete Data via EM Algorithm," *J. Royal Stat. Soc*, B 39, pp 1-38, 1977

4 R.J. Hathaway, "A Constrained EM Algorithm for Univariate Normal Mixtures," *J. Statist. Comput. Simul.*, Vol. 23, pp 211-230, 1986

5 A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall Inc., New Jersey:, 1988

6 J. Rissanen, "A Universal Prior For Integers And Estimation By Minimum Description Length," *Ann. Statist.*, Vol 11, No 2, pp 416-431, 1983

7 A.R. Barron, "Minimum Complexity Density Estimation," *IEEE Trans. Info. Theory*, Vol 37, No 4, pp1034-1054, 1991

8 J. Sheinvald, B. Dom, W. Niblack, D. Steele, "Unsupervised Image Segmentation using the Minimum Description Length Principle," *IBM Research Report*, RJ 8487 (76695), November 1991

9 K. Fukunaga, D. Kessell, "Nonparametric Bayes Error Estimation Using Unclassified Samples," *IEEE Trans. Info. Theory*, Vol. IT-19, pp 434-440, 1973

10 D.S. Moore, S.J. Whitsitt, D.A. Landgrebe, "Variance Comparisons for Unbiased Estimators of Probability of Correct Classification," *IEEE Trans. Info. Theory*, 22, pp 102-105, 1976

11 K.E. Basford, G.J. McLachlan, "Estimation of Allocation Rates in a Cluster Analysis Context," *J.American Stat. Assoc.*, Vol 80, pp 286-293, 1985